How to Profile OPeNDAP Server

Use this procedure to profile the memory usage on an OPeNDAP server.

We will be using the valgrind tool with the massif memory profiler. You may need root or administrative privileges to start/stop the OPeNDAP server. You might want to create a new OPeNDAP instance or ensure that others won't attempt to use the server during profiling.

Massif provides a wealth of information, showing the max memory used, as well as how memory usage changes over time. It includes several detailed "snapshots" of memory usage – showing exactly where in the code memory was allocated.

Step-by-step guide

Run Valgrind to capture profile data

1. Setup environment:
    a. `% source spath.sh`

2. Stop the server (if currently running):
    a. `./besctl stop`

3. Start the server using valgrind + massif. Trace children is required because server-side functions run in a forked executable.Output will be captured in the file "valmem.txt" and multiple "massif.out.[pid]" files.
    a. `valgrind --tool=massif --trace-children=yes ./besctl start > valmem.txt 2>&1`

4. Run a test that hits this OPeNDAP server. Make sure to keep it on the small side since it will run much more slowly than normal
5. Stop the server again, and restart it as normal if desired
    a. `./besctl stop`

Find the file containing your test data

You should see a bunch of "massif.out.[pid]" files. We need to find which pid ran the server-side function. The best way is to look for the string "miic" in the output. This will find the file that ran the beslistener process that ran the miic plugin. You can find it via grep:

```
grep miic2d_avg_histo massif.out.*
```

There are normally two beslisteners started. However, if you only ran one request, only one will have the information you need.

Analyze the data

The massif.out files are not by themselves easily analyzed. The command ms_print will prepare the file for analysis.

```
ms_print massif.out.[pid] > formatted_output
```

The file will show overall memory allocation behavior as a graph, and also detailed memory allocation requests.

Massif generated memory allocation graph

```
MB
173.4^                                              #
     |                                              #
     |                                              #
     |                                              #
     |                                              #
     |                                              #
     |                                              #            :::
     |                                   ::@::::::::#            :
:::::::::::
     |                          :        ::@::::    #    ::  ::::: :
:
     |                          :    ::  ::@::::    # :::: :::: :: :
:
     |                          :  ::: :@::@::::    #::  : :::: :: :
:
     |                          :::: : :@::@::::    #::  : :::: :: :
:
     |                          :: : : :@::@::::    #::  : :::: :: :
:
     |                          :: : : :@::@::::    #::  : :::: :: :
:
     |                :@:@:@@@@@:: : : :@::@::::    #::  : :::: :: :
:
     |                :@:@:@    :: : : :@::@::::    #::  : :::: :: :
:
     |                :@:@:@    :: : : :@::@::::    #::  : :::: :: :
@
     |            :::::@:@:@    :: : : :@::@::::    #::  : :::: :: :
@
     |       ::: ::::::@:@:@    :: : : :@::@::::    #::  : :::: :: :
@
     |       ::::::::::@:@:@    :: : : :@::@::::    #::  : :::: :: :
@
   0
+----------------------------------------------------------------------->Gi
     0
10.32
```

In the charts, the X axis is giga-instructions and the Y axis is snapshot memory usage.

How to interpret the graph:

- Each vertical line of characters represents one snapshot.
- The high-water mark of memory usage is shown as "#", 173.4 MB.
- The vertical lines shown as "@" (and the peak) are detailed snapshots. Massif shows, by percent, where all the memory in that snapshot was allocated in your code.

Related articles

- How to setup & run a MIIC OPeNDAP Server
- How to build the MIIC OPeNDAP plugin
- How to debug OPeNDAP SSFs
- How to Profile OPeNDAP Server
- How to plot OPeNDAP requests per minute